

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6817523号
(P6817523)

(45) 発行日 令和3年1月20日(2021.1.20)

(24) 登録日 令和3年1月4日(2021.1.4)

(51) Int.Cl.

F 1

G06F 8/38 (2018.01)
G06F 3/0481 (2013.01)G06F 8/38
G06F 3/0481

請求項の数 8 (全 21 頁)

(21) 出願番号 特願2019-96621 (P2019-96621)
 (22) 出願日 令和1年5月23日 (2019.5.23)
 (62) 分割の表示 特願2018-89328 (P2018-89328)
 分割
 原出願日 平成25年9月26日 (2013.9.26)
 (65) 公開番号 特開2019-153340 (P2019-153340A)
 (43) 公開日 令和1年9月12日 (2019.9.12)
 審査請求日 令和1年6月14日 (2019.6.14)

(73) 特許権者 390002761
 キヤノンマーケティングジャパン株式会社
 東京都港区港南2丁目16番6号
 (73) 特許権者 592135203
 キヤノンITソリューションズ株式会社
 東京都港区港南2丁目16番6号
 (74) 代理人 100189751
 弁理士 木村 友輔
 (72) 発明者 柴本 文洋
 東京都品川区東品川2丁目4番11号 キ
 ヤノンソフトウェア株式会社内
 (72) 発明者 上田 熊
 東京都品川区東品川2丁目4番11号 キ
 ヤノンソフトウェア株式会社内

最終頁に続く

(54) 【発明の名称】情報処理装置、情報処理装置の制御方法、およびプログラム

(57) 【特許請求の範囲】

【請求項 1】

情報処理装置を、

サーバで実行可能なプログラムを生成するための定義情報であって、前記プログラムが実行されることにより生成される画面情報に含まれる画面項目に係る定義情報を用いて、前記画面情報を生成するためのプログラムを生成するプログラム生成手段と、

前記定義情報のうち前記画面項目の表示に係る定義情報を抽出する表示定義情報抽出手段と、

前記プログラムが実行されることにより生成される画面情報に対応するプレビュー画面を、前記表示定義情報抽出手段により抽出される前記画面項目の表示に係る定義情報を用いて生成するプレビュー画面生成手段

として機能させるためのプログラム。

【請求項 2】

前記表示定義情報抽出手段は、前記定義情報のデータ種別を抽出し、

前記プレビュー画面生成手段は、前記表示定義情報抽出手段により抽出されたデータ種別に基づいたオブジェクトを配置することを特徴とする請求項1に記載のプログラム。

【請求項 3】

前記定義情報は、ユーザからの入力に従って変更可能であって、

前記プレビュー画面生成手段は、

前記ユーザからの入力に従って変更された定義情報を用いて、前記プレビュー画面を生成すること
を特徴とする請求項 1 又は 2 に記載のプログラム。

【請求項 4】

前記プレビュー画面に含まれる画面項目に対して設定する値を記憶する値記憶手段として更に機能させ、

前記プレビュー画面生成手段は、

前記値記憶手段により記憶されている値が設定された前記画面項目を含む前記プレビュー画面を生成すること

を特徴とする請求項 1 乃至 3 のいずれか 1 項に記載のプログラム。 10

【請求項 5】

前記プレビュー画面生成手段は、

前記値記憶手段に記憶されている値に基づき、前記値記憶手段に記憶されている値を設定する画面項目に対しては、前記値記憶手段に記憶されている当該画面項目に対応する値を設定し、一方、前記値記憶手段に記憶されている値を設定しない画面項目に対しては、当該画面項目に対応するデフォルト値を設定し、プレビュー画面を生成すること
を特徴とする請求項 4 に記載のプログラム。

【請求項 6】

前記プレビュー画面生成手段により生成されるプレビュー画面を閲覧するエンドユーザのロール又は表示言語の少なくともいずれか 1 つに応じて表示する項目を制御するプレビュー画面制御手段
として更に機能させ、

前記値記憶手段は、エンドユーザのロール又は表示言語の少なくともいずれか 1 つを記憶し、

前記プレビュー画面生成手段は、

前記値記憶手段により記憶されているエンドユーザのロール又は表示言語の少なくともいずれか 1 つを用いて、前記プレビュー画面制御手段に応じて項目の表示を制御したプレビュー画面を生成すること

を特徴とする請求項 4 又は 5 のいずれか 1 項に記載のプログラム。 20

【請求項 7】

情報処理装置が、

サーバで実行可能なプログラムを生成するための定義情報をあって、前記プログラムが実行されることにより生成される画面情報に含まれる画面項目に係る定義情報を用いて、前記画面情報を生成するためのプログラムを生成するプログラム生成ステップと、

前記定義情報をうち前記画面項目の表示に係る定義情報を抽出する表示定義情報抽出ステップと、

前記プログラムが実行されることにより生成される画面情報に対応するプレビュー画面を、前記表示定義情報抽出ステップにより抽出される前記画面項目の表示に係る定義情報を用いて生成するプレビュー画面生成ステップと、
を実行することを特徴とする処理方法。 40

【請求項 8】

サーバで実行可能なプログラムを生成するための定義情報をあって、前記プログラムが実行されることにより生成される画面情報に含まれる画面項目に係る定義情報を用いて、前記画面情報を生成するためのプログラムを生成するプログラム生成手段と、

前記定義情報をうち前記画面項目の表示に係る定義情報を抽出する表示定義情報抽出手段と、

前記プログラムが実行されることにより生成される画面情報に対応するプレビュー画面を、前記表示定義情報抽出手段により抽出される前記画面項目の表示に係る定義情報を用いて生成するプレビュー画面生成手段と、
を備えることを特徴とする情報処理装置。 50

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、プログラム自動生成装置を用いた開発であって、ビジネスロジック、データベースモデルなどのそれぞれのモジュールの開発が未完成であっても、確認したい画面を短時間で生成することを可能とする技術に関する。

【背景技術】**【0002】**

従来から、ウェブアプリケーションの開発を支援するため、ユーザはG U Iレイアウト(プログラム実行時にH T M Lとして表示される部分)や、ビジネスロジック、データベースモデルを自動生成するツール(製品)がある。これらの開発支援ツールを用いて、ウェブアプリケーションが表示する画面を作成する場合、入力フィールドやボタンなどの部品について、ツール(製品)毎に決められた定義を行う必要がある。画面作成を開発者などが容易に行うため、アプリケーションが表示するH T M Lをグラフィカルに編集するエディタを搭載するツールもある。

【0003】

しかしながら、従来のシステムにおいて、開発中(プログラム自動生成ツールでの規約に基づいたアプリケーションの定義)に最終的に生成されるH T M Lを確認しようとすると、プログラム自動生成ツールとして、画面(H T M L)と、ビジネスロジックやデータベースモデルなどを分割定義し、後で関連付けるようなツールを使用する必要があった。

【0004】

特許文献1には、表示項目などのオブジェクトであるフィールドと、ビヘイビアと呼ばれるフィールドにおけるイベントをアクション実行やエラーチェックとしてあわせて定義することによりウェブ画面を生成する、ウェブアプリケーションのオーサリングツールに関する技術が記載されている。オーサリングツールは、フィールドの配置とビヘイビアの定義をするための編集ビューを持ち、インスタンス化することでH T M L、スクリプトのコードを生成し、動的なウェブページを生成可能とする。

【先行技術文献】**【特許文献】****【0005】**

【特許文献1】特開2003-99257号公報

【発明の概要】**【発明が解決しようとする課題】****【0006】**

しかしながら、特許文献1の技術においては、開発の途中段階で画面(H T M L)の状態を確認できない場合がある。すなわち、画面(H T M L)だけを先に開発することで、未完成なビヘイビアがない状態であるか、あるいは作成しかけたビヘイビアは、作成を完了させておく必要がある。すなわち、画面上の項目と関連付けられたイベントは、(最終的な完成物ではないにしても)一定レベルの動作を保証されていなければ、画面(H T M L)の状態を確認することができない。

【0007】

従って、画面(H T M L)と、ビヘイビアと連携して開発しながらも、それぞれの連携が完全に取れていない段階では、画面(H T M L)だけを確認できないという問題がある。

【0008】

また、そもそも特許文献1に記載のオーサリングツールは、クライアント(ウェブブラウザ)で動的に動作するためのH T M Lを作成するものであるため、アプリケーションサーバとの連係で動作するビジネスロジックや、データモデルを含むスケールのウェブアプリケーションを考えたとき、ビジネスロジックのコンパイルや、データモデルの生成など

10

20

30

40

50

、画面（H T M L）に関するプログラムを生成した後でないと、画面（H T M L）の状態を確認することはできることになる。これでは、開発効率が低下するという問題を引き起こすことになる。

【0009】

本発明の目的は、上記の問題に鑑み、画面項目の表示に係る定義情報を用いてプレビュー画面を生成してアプリケーションの開発効率を向上させることが可能なプログラム自動生成装置の技術を提供することである。

【課題を解決するための手段】

【0010】

本発明は、情報処理装置を、サーバで実行可能なプログラムを生成するための定義情報であって、前記プログラムが実行されることにより生成される画面情報に含まれる画面項目に係る定義情報を用いて、前記画面情報を生成するためのプログラムを生成するプログラム生成手段と、前記定義情報のうち前記画面項目の表示に係る定義情報を抽出する表示定義情報抽出手段と、前記プログラムが実行されることにより生成される画面情報に対応するプレビュー画面を、前記表示定義情報抽出手段により抽出される前記画面項目の表示に係る定義情報を用いて生成するプレビュー画面生成手段として機能させることを特徴とする。

【発明の効果】

【0011】

本発明により、画面項目の表示に係る定義情報を用いてプレビュー画面を生成してアプリケーションの開発効率を向上させることが可能となる。

【図面の簡単な説明】

【0012】

【図1】本発明の実施形態に係わる情報処理装置の基本的な機能構成、および各機能の呼び出し関係を示す図である。

【図2】本発明の実施形態に係わる情報処理装置のハードウェア構成の一例を示すプロック図である。

【図3】本発明の実施形態に係わるプログラム自動生成ツールのうちG U Iレイアウトの定義を行う画面構成の一例である。

【図4】本発明の実施形態に係わるプログラム自動生成ツールのうち項目の定義を行う画面構成の一例である。

【図5】本発明の実施形態に係わるプログラム自動生成ツールで定義された項目一覧の情報を記憶する項目一覧モデル記憶部におけるデータ構成の一例を示す図である。

【図6】本発明の実施形態に係わるG U Iレイアウトの定義を項目一覧の定義に同期するためのシーケンス図の一例である。

【図7】本発明の実施形態に係わる項目一覧の定義をG U Iレイアウトの定義に同期するためのシーケンス図の一例である。

【図8】本発明の実施形態に係わる情報処理装置のプレビューに関する機能構成、および各機能の呼び出し関係を示す図の一例である。

【図9】本発明の実施形態に係わるプレビューH T M Lのプレビュー状態を示す図の一例である。

【図10】本発明の実施形態に係わるプレビューH T M Lのファイル保管場所（フォルダ）を指定するユーザ画面を示す図の一例である。

【図11】本発明の実施形態に係わるプレビューH T M Lを生成するために必要な定義情報をアプリケーション定義から取得した結果のデータ構成を示す図の一例である。

【図12】本発明の実施形態に係わるプレビューH T M Lの生成処理を示すフローチャートの一例である。

10

20

30

40

50

【発明を実施するための形態】**【0013】**

以下、本発明の実施の形態を、図面を参照して詳細に説明する。まず、図1～図7を用いて、HTMLプレビュー生成の前提となるGUIレイアウトについて、説明する。

【0014】

図1は、本発明の実施形態に係わる情報処理装置100の機能構成、および各機能の呼び出し関係を示す図である。図1において、ユーザ操作(101)は、GUI(グラフィカル・ユーザ・インターフェース)における定義画面(プログラム自動生成ツールにおいてプログラムの定義を行う画面)から行われる。具体的には、情報処理装置100にて動作しているプログラム自動生成ソフトにおけるGUIレイアウトタブ(図3)、項目一覧タブ(図4)でGUIレイアウト、または項目一覧の定義を行うが、タブの相互切り替えに際しての処理に関連するソフトウェア構成部品と、構成部品間の要求の図である。なお、GUIレイアウトタブおよび項目一覧タブに定義されている情報は、定義画面ヘッダの定義ファイル名(図3、図4の画面例では“USERREQ.wpx”)、あるいは当該画面を一意的に決定する情報により特定される。

【0015】

ユーザ操作(101)により、項目一覧タブ選択要求121が、入出力定義エディタコントローラ部102に通知されると、入出力定義エディタコントローラ部102は、項目一覧モデル変換部103にレイアウトモデル変換要求122を通知する。

【0016】

項目一覧モデル変換部103は、GUIレイアウト定義を項目一覧定義に変換するため、現在のGUIレイアウト定義(GUIレイアウトタブで定義されたレイアウト定義でありレイアウト部品情報の定義の一覧)をレイアウトモデル記憶部107から取得する(レイアウトモデル変換時取得要求123)。さらに、項目一覧モデル変換部103は、取得した現在のGUIレイアウトを、項目一覧定義(項目情報の定義の一覧)に変換し、変換結果を項目一覧モデル記憶部104に登録する(既にある場合は更新する)。項目一覧モデル更新要求124)。

【0017】

入出力定義エディタコントローラ部102は、項目一覧定義の登録または更新後、当該登録または更新結果を取得する(項目一覧モデル表示時取得要求125)。入出力定義エディタコントローラ部102は、取得した項目一覧定義を項目一覧タブに表示するよう項目一覧表示部105に要求し、項目一覧表示部105は要求に従って項目一覧定義の表示を行う。

【0018】

ユーザ操作(101)により、レイアウトタブ選択要求141が、入出力定義エディタコントローラ部102に通知されると、入出力定義エディタコントローラ部102は、GUIレイアウトモデル変換部106に項目一覧モデル変換要求142を通知する。

【0019】

GUIレイアウトモデル変換部106は、項目一覧定義をGUIレイアウトモデル定義に変換するため、現在の項目一覧定義を項目一覧モデル記憶部104から取得する(項目一覧モデル変換時取得要求143)。さらに、GUIレイアウトモデル変換部106は、取得した現在の項目一覧定義を、GUIレイアウト定義に変換し、レイアウトモデル記憶部107に登録する(既にある場合は更新する)。レイアウトモデル更新要求144)。

【0020】

入出力定義エディタコントローラ部102は、GUIレイアウト定義の登録または更新後、当該登録または更新結果を取得する(レイアウトモデル表示時取得要求145)。入出力定義エディタコントローラ部102は、取得したGUIレイアウト定義をGUIレイアウトタブに描画するための準備としてGUIレイアウトコントローラ部108にGUIレイアウト部品の生成を要求する(GUIレイアウト生成要求146)。

【0021】

10

20

30

40

50

G U I レイアウト生成要求を受けた G U I レイアウトコントローラ部 1 0 8 は、レイアウトモデル記憶部 1 0 7 から取得した G U I レイアウト定義に基づき、描画するための G U I 部品などを生成する。まず、G U I レイアウト上の G U I 部品（テキスト入力ボックスやボタンなど）を配置するためのベースとなる部品（ベース部品）を生成し、G U I レイアウト記憶部 1 0 9（G U I レイアウト & 部品記憶部）に追加する（G U I レイアウト追加要求 1 4 7）。

【 0 0 2 2 】

G U I レイアウトコントローラ部 1 0 8 は、さらにベース部品に配置する G U I 部品を生成するため、ベース部品の直下の G U I 部品ごとに、G U I 部品生成要求 1 4 8 を G U I 部品コントローラ部 1 1 0 に発行する。

10

【 0 0 2 3 】

G U I 部品コントローラ部 1 1 0 は、G U I 部品生成要求 1 4 8 により要求された G U I 部品を、G U I レイアウト定義の個々の部品の定義（項目情報から変換された定義）に基づき生成する。G U I レイアウト定義において、G U I 部品に入れ子の G U I 部品（ベース部品の直下に配置される部品ではなく、G U I 部品上に配置される子供となる G U I 部品）が定義されている場合には、入れ子の部品を生成するように G U I 部品コントローラ部 1 1 0 に再帰的に要求を行う（G U I 部品生成要求 1 4 9）。生成した G U I 部品は、G U I レイアウト記憶部 1 0 9 に追加する（G U I 部品追加要求 1 5 0）。

【 0 0 2 4 】

G U I レイアウトコントローラ部 1 0 8 は、G U I 部品（ベース部品を含む）の生成が完了すると、G U I レイアウト描画部 1 1 1 に G U I レイアウト描画要求 1 5 1 を発行する。この要求により、G U I レイアウト描画部 1 1 1 は、G U I レイアウトタブに G U I レイアウトの表示を行う。

20

【 0 0 2 5 】

図 2 は、本発明の実施形態に係わる情報処理装置 1 0 0（プログラム自動生成装置）のハードウェア構成の一例を示すブロック図である。

【 0 0 2 6 】

図 2 に示すように、情報処理装置 1 0 0 は、システムバス 2 0 4 を介して C P U（C e n t r a l P r o c e s s i n g U n i t）2 0 1、R A M（R a n d o m A c c e s s M e m o r y）2 0 2、R O M（R e a d O n l y M e m o r y）2 0 3、入力コントローラ 2 0 5、ビデオコントローラ 2 0 6、メモリコントローラ 2 0 7、通信 I / F コントローラ 2 0 8 等が接続された構成を探る。C P U 2 0 1 は、システムバス 2 0 4 に接続される各デバイスやコントローラを統括的に制御する。

30

【 0 0 2 7 】

また、R O M 2 0 3 あるいは外部メモリ 2 1 1 には、C P U 2 0 1 の制御プログラムである B I O S（B a s i c I n p u t / O u t p u t S y s t e m）や O S（O p e r a t i n g S y s t e m）や、各サーバあるいは各 P C が実行する機能を実現するために必要な後述する各種プログラム等が記憶されている。また、本発明を実施するために必要な情報が記憶されている。なお外部メモリはデータベースであってもよい。

【 0 0 2 8 】

R A M 2 0 2 は、C P U 2 0 1 の主メモリ、ワークエリア等として機能する。C P U 2 0 1 は、処理の実行に際して必要なプログラム等を R O M 2 0 3 あるいは外部メモリ 2 1 1 から R A M 2 0 2 にロードし、ロードしたプログラムを実行することで各種動作を実現する。

40

【 0 0 2 9 】

また、入力コントローラ 2 0 5 は、キーボード（K B）2 0 9 や不図示のマウス等のポインティングデバイス等からの入力を制御する。

【 0 0 3 0 】

ビデオコントローラ 2 0 6 は、ディスプレイ 2 1 0 等の表示器への表示を制御する。尚、表示器は液晶ディスプレイ等の表示器でもよい。これらは、必要に応じて管理者が使用

50

する。

【0031】

メモリコントローラ207は、ポートプログラム、各種のアプリケーション、フォントデータ、ユーザファイル、編集ファイル、各種データ等を記憶する外部記憶装置（ハードディスク（HD））や、フレキシブルディスク（FD）、あるいは、PCMCIA（Personal Computer Memory Card International Association）カードスロットにアダプタを介して接続されるコンパクトフラッシュ（登録商標）メモリ等の外部メモリ211へのアクセスを制御する。

【0032】

通信I/Fコントローラ208は、ネットワークを介して外部機器と接続・通信し、ネットワークでの通信制御処理を実行する。例えば、TCP/IP（Transmission Control Protocol/Internet Protocol）を用いた通信等が可能である。10

【0033】

尚、CPU201は、例えばRAM202内の表示情報用領域へアウトラインフォントの展開（ラスタライズ）処理を実行することにより、ディスプレイ210上に表示することが可能である。また、CPU201は、ディスプレイ210上のマウスカーソル（図示しない）等によるユーザ指示を可能とする。

【0034】

本発明を実現するための後述する各種プログラムは、外部メモリ211に記録されており、必要に応じてRAM202にロードされることによりCPU201によって実行されるものである。さらに、上記プログラムの実行時に用いられる定義ファイルおよび各種情報テーブル等も、外部メモリ211に格納されており、これらについての詳細な説明についても後述する。20

【0035】

図3は、本発明の実施形態に係わるプログラム自動生成ツールのうちGUIレイアウトの定義を行う画面構成の一例である。GUIレイアウト定義画面は、主に次のエリアから構成されている。すなわち、定義された単位毎にフォルダ階層としてデータが格納される定義エクスプローラ領域301、1つのGUIレイアウト定義を視覚的に配置操作するためのキャンバス領域302、キャンバス領域302に配置するGUI部品（例えばテキストフィールド304）を選択するパレット領域303から構成されている。30

【0036】

また、開発者が様々な定義情報を記述するため、プログラム自動生成ツールの定義画面で共通に表示されるタブが配置されている。例えば、項目一覧タブ305、レイアウトタブ306であり、これらは図4の項目一覧定義画面でも表示される。本発明の実施形態では、GUIレイアウト定義画面（図3）と項目一覧定義画面（図4）を切り替えながら操作する。

【0037】

また、本発明の実施形態では、定義エクスプローラ領域301にあるデータモデル308をパレット領域303のGUI部品と同様に、キャンバス領域302に配置することが可能である。すなわち、データモデル308から、データベースのテーブルに記載されたデータ項目（カラム）を1つまたは複数選択し、キャンバス領域302に配置する。そうすると、データベースのスキーマとして定義されたデータ項目（カラム）の情報（カラム名、データ種別、サイズなど）を抽出して、例えば、データ種別がテキストや数値である場合にはテキストフィールド、booleanである場合にはチェックボックスなどとして配置することができる。40

【0038】

GUIレイアウト定義画面300bは、GUIレイアウト定義画面300aに対してテキストフィールド307を1つ追加し、名前を“ユーザID”と設定した場合を示している。タイトルや入力領域のサイズなどは、キャンバス領域302においてマウスでクリッ

クして更新可能状態として交信してもよいし、また G U I 部品をクリックする際に、G U I レイアウト定義画面 3 0 0 の左下の領域（“Property”定義領域）に、属性として G U I 部品に関連する情報の一覧を表示し、属性をキーボード入力により修正することも可能である。

【0039】

図 4 は、本発明の実施形態に係わるプログラム自動生成ツールのうち項目一覧の定義を行う画面構成の一例である。項目一覧定義画面は、主張な領域は、項目一覧設定領域 4 0 1 である。

【0040】

項目一覧設定領域 4 0 1 は表形式になっており、開発者は 1 行に 1 つの情報（例えば G U I レイアウトにおけるテキストフィールド、チェックボックス、ボタン、テキスト）などを詳細の属性と共に定義する。

【0041】

項目一覧定義画面 4 0 0 b は、項目一覧定義画面 4 0 0 a に対して項目タイプ“入力”を 1 つ追加し、名前を“ユーザ ID”と設定した場合を示している（行 4 0 2 の定義）。行 4 0 2 の定義は、図 3 のテキストフィールド 3 0 7 に対応するものである。

【0042】

ここで、図 3 と図 4 の定義の関係を説明する。例えば図 3 の G U I レイアウト定義画面 3 0 0 a を表示しているときに、項目一覧タブ 3 0 5 を選択すると、表示は図 4 の項目一覧定義画面 4 0 0 a に切り替えられる。実質的に開発しているプログラムの同じ情報を定義している。図 4 において、前述の通り行 4 0 2 を追加して項目一覧定義画面 4 0 0 b とした後、レイアウトタブ 3 0 6 を選択すると、G U I レイアウト定義画面 3 0 0 b のようにレイアウト上でもテキストフィールド 3 0 7 が追加されて描画される。

【0043】

逆に、G U I レイアウト定義画面 3 0 0 a に対してレイアウト上でもテキストフィールド 3 0 7 を追加して G U I レイアウト定義画面 3 0 0 b とした後、項目一覧タブ 3 0 5 を選択すると、項目一覧定義画面 4 0 0 b のように項目一覧上でも行 4 0 2 の定義が追加される。

【0044】

すなわち、G U I レイアウト定義と項目一覧定義は、タブ（項目一覧タブ 3 0 5 およびレイアウトタブ 3 0 6）の操作をトリガとして、同期がとられている。

【0045】

さらに、後述する図 5 の項目一覧定義を記憶するデータ構成 rows [i] と、不図示の G U I レイアウト定義を記憶するデータ構成には、rowIndex として各定義の定義順が記載されている。そのため、G U I レイアウト定義で最上部に定義された G U I 部品に対応する項目の定義は、項目一覧定義の 1 行目に表示される。これにより、定義の位置関係についても同期がとられ、いずれの定義を変更してもプログラムロジックと G U I レイアウトの関係が壊れてしまうことはない。

【0046】

以上により、図 3 と図 4 の説明を完了する。このように、項目一覧における開発と G U I レイアウトにおける開発（すなわち項目の定義も部品の定義）において同期がとられることで、開発中のどのタイミングでも G U I レイアウトを確認可能とし、またプログラムロジックと G U I レイアウトのいずれを編集したとしても対応関係を修正する必要をなくすることにより、開発効率を向上させることが可能となるという効果を得ることができる。

【0047】

図 5 は、本発明の実施形態に係わるプログラム自動生成ツールで定義された項目一覧の情報を記憶する項目一覧モデル記憶部におけるデータ構成の一例を示す図である。記載したデータは、図 4 の項目一覧定義画面 4 0 0 の項目一覧設定領域 4 0 1 において定義した行のうち、例として 5 行を記載したものである。この情報は、RAM202 上に一時的に

10

20

30

40

50

記憶されていてもよいし、データベースに記憶してもよい。以下、RAM202に記憶されているイメージで説明する。

【0048】

項目一覧定義は配列rowsの各項目定義は、rows[0]～rows[4]に格納されている。rows[i](各要素)は、図5示すようにdataという要素を持ち、複数の属性値、例えば入力欄を示す“I”、項目IDを示す“USERID”、項目の名前を示す“ユーザID”などが設定される。

【0049】

さらに、プログラム自動生成に使用される各種属性が定義されている。また、“owP
ropertyList”には、名前を表示する領域のサイズ“labelWidth”
、テキスト入力欄の領域サイズ“fieldWidth”などが指定される。
10

【0050】

以上で、項目一覧モデル記憶部104におけるデータ構成の説明を完了する。さらに、不図示ではあるが、図3のGUIレイアウトについても、レイアウトモデル記憶部107に記憶される。両者のデータ構成はほぼ同一のものである。本発明の実施形態では、項目一覧モデル記憶部104に記憶された情報を中心とするため、レイアウトモデル記憶部107におけるデータ構成は、各要素(レイアウト部品情報)が、対応する項目情報へのリンクを有すること、また項目情報(rows[i])における領域サイズの情報は、HTMLに変換した際のサイズを示すが、レイアウト部品譲歩においては、GUIレイアウト定義画面300のキャンバス領域302に表示する際に適したサイズに変換されている、など一部の情報が異なる。項目一覧モデル記憶部104、レイアウトモデル記憶部107のデータ構成はあくまで一例であって、他の構成の形式で定義されていてもよい。
20

【0051】

図6は、本発明の実施形態に係わるGUIレイアウトの定義を項目一覧の定義に同期するためのシーケンス図の一例である。図6のシーケンス図の各ステップは、情報処理装置100のCPU201により実行される。

【0052】

S601において、ユーザ操作(101)により項目一覧タブが選択された際に、プログラム開発ツールから項目一覧タブ選択の要求が発行され、入出力定義エディタコントローラ部102が、定義画面を項目一覧タブ(図4)に切り替えるための要求を受け付ける(項目一覧タブ選択要求121)。
30

【0053】

S602において、入出力定義エディタコントローラ部102が、GUIレイアウト定義を項目一覧定義に変換するように、項目一覧モデル変換部103に要求する(レイアウトモデル変換要求122)。ここで、いずれの定義に対する変換を要求するかは、図1で説明したとおり定義を一意的に決定する情報により特定される。

【0054】

S603において、項目一覧モデル変換部103が、GUIレイアウトの定義をレイアウトモデル記憶部107から取得する(レイアウトモデル変換時取得要求123)。
40

【0055】

S604において、項目一覧モデル変換部103は、更に取得したGUIレイアウトを項目一覧定義に変換する。前述したように、GUIレイアウト定義では、GUIレイアウト定義画面300のキャンバス領域302に表示する際に適したサイズを保持しているため、それらのサイズをプログラム自動生成時に、HTMLの部品として生成するために適切なサイズに変換する、などの変換処理を行う。

【0056】

S605において、項目一覧モデル変換部103は、取得した現在のGUIレイアウトを、項目一覧定義(項目情報の定義の一覧)に変換し、変換結果を項目一覧モデル記憶部104に登録する(既にある場合は更新する。項目一覧モデル更新要求124)。

【0057】

10

20

30

40

50

S 6 0 6において、入出力定義エディタコントローラ部102は、登録または更新された項目一覧定義を、項目一覧モデル記憶部104から取得する（項目一覧モデル表示時取得要求125）。

【0058】

S 6 0 7において、入出力定義エディタコントローラ部102は、取得した項目一覧定義を項目一覧タブに表示するよう項目一覧表示部105に要求し、項目一覧表示部105は要求に従って項目一覧定義の表示を行う。

【0059】

図7は、本発明の実施形態に係わる項目一覧の定義をG U Iレイアウトの定義に同期するためのシーケンス図の一例である。図6のシーケンス図の各ステップは、情報処理装置100のC P U 2 0 1により実行される。
10

【0060】

S 7 0 1において、ユーザ操作（101）によりレイアウトタブが選択された際に、プログラム開発ツールからレイアウトタブ選択の要求が発行され、入出力定義エディタコントローラ部102が、定義画面をレイアウト定義タブ（図3）に切り替えるための要求を受け付ける（レイアウトタブ選択要求141）。

【0061】

S 7 0 2において、入出力定義エディタコントローラ部102は、G U Iレイアウトモデル変換部106に項目一覧モデル変換要求142を通知する（項目一覧モデル変換要求142）。ここで、いずれの定義に対する変換を要求するかは、図1で説明したとおり定義を一意的に決定する情報により特定される。
20

【0062】

S 7 0 3において、G U Iレイアウトモデル変換部106は、項目一覧定義をG U Iレイアウトモデル定義に変換するため、現在の項目一覧定義を項目一覧モデル記憶部104から取得する（項目一覧モデル変換時取得要求143）。

【0063】

S 7 0 4において、G U Iレイアウトモデル変換部106は、取得した現在の項目一覧定義を、G U Iレイアウト定義に変換する。

【0064】

S 7 0 5において、G U Iレイアウトモデル変換部106は、変換したG U Iレイアウト定義をレイアウトモデル記憶部107に登録する（既にある場合は更新する。レイアウトモデル更新要求144）。
30

【0065】

S 7 0 6において、入出力定義エディタコントローラ部102は、G U Iレイアウト定義の登録または更新後、当該登録または更新結果を取得する（レイアウトモデル表示時取得要求145）。

【0066】

S 7 0 7において、入出力定義エディタコントローラ部102は、取得したG U Iレイアウト定義をG U Iレイアウトタブに描画するための準備としてG U Iレイアウトコントローラ部108にG U Iレイアウト部品の生成を要求する（G U Iレイアウト生成要求146）。
40

【0067】

S 7 0 8において、G U Iレイアウト生成要求を受けたG U Iレイアウトコントローラ部108は、レイアウトモデル記憶部107から取得したG U Iレイアウト定義に基づき、描画するためのG U I部品などを生成する。まず、G U Iレイアウト上のG U I部品（テキスト入力ボックスやボタンなど）を配置するためのベースとなる部品（ベース部品）を生成し、G U Iレイアウト記憶部109（G U Iレイアウト＆部品記憶部）に追加する（G U Iレイアウト追加要求147）。

【0068】

S 7 0 9において、G U Iレイアウトコントローラ部108は、さらにベース部品に配
50

置する G U I 部品を生成するため、ベース部品の直下の G U I 部品ごとに、G U I 部品生成要求 1 4 8 を G U I 部品コントローラ部 1 1 0 に発行する。

【 0 0 6 9 】

S 7 1 0において、G U I 部品コントローラ部 1 1 0は、G U I 部品生成要求 1 4 8により要求されたG U I 部品を、G U I レイアウト定義の個々の部品の定義（項目情報から変換された定義）に基づき生成する。G U I レイアウト定義において、G U I 部品に入れ子のG U I 部品（ベース部品の直下に配置される部品ではなく、G U I 部品上に配置される子供となるG U I 部品）が定義されている場合には、入れ子の部品を生成するようにG U I 部品コントローラ部 1 1 0に再帰的に要求を行う（G U I 部品生成要求 1 4 9）。生成したG U I 部品は、G U I レイアウト記憶部 1 0 9に追加する（G U I 部品追加要求 1 5 0）。 10

【 0 0 7 0 】

S 7 1 1において、G U I レイアウトコントローラ部 1 0 8は、G U I 部品（ベース部品を含む）の生成が完了すると、G U I レイアウト描画部 1 1 1にG U I レイアウト描画要求 1 5 1を発行する。この要求により、G U I レイアウト描画部 1 1 1は、G U I レイアウトタブにG U I レイアウトの表示を行う。

【 0 0 7 1 】

図 6 と図 7 のシーケンス図で説明したとおり、項目一覧タブとレイアウトタブを切り替えるユーザの操作をトリガとして、両者に表示されるデータ（項目一覧定義とG U I レイアウト定義）が再生成され、項目一覧定義（プログラムロジック）と、G U I レイアウト定義（H T M L 生成などのための画面情報）を、手動で対応付けるための操作や、対応情報の作成などを行う必要がなくなる。これにより、プログラム自動生成ツールによるプログラム定義の作業の生産性が向上するという効果が得られる。 20

【 0 0 7 2 】

次に、図 8 ~ 図 1 2 を用いて、プレビュー H T M L 生成についての詳細な説明を行う。プレビュー H T M L は、アプリケーションにおいて、ビジネスロジック（コントローラ）、データベースモデルに対応するビューである H T M L を、プレビューとして開発途中に生成し確認するための H T M L 画面である。

【 0 0 7 3 】

図 8 は、本発明の実施形態に係わる情報処理装置のプレビューに関する機能構成、および各機能の呼び出し関係を示す図の一例である。また、図 8 では、各機能部や要求の概略だけを説明し、詳細は図 1 2 のフローチャートで説明する。 30

【 0 0 7 4 】

まず、ユーザ操作（1 0 1）は、図 1 で説明したとおり、G U I（グラフィカル・ユーザー・インターフェース）における定義画面（プログラム自動生成ツールにおいてプログラムの定義を行う画面）から行われる。

【 0 0 7 5 】

具体的には、情報処理装置 1 0 0 にて動作しているプログラム自動生成ソフトにおけるG U I レイアウトタブ（図 3）、項目一覧タブ（図 4）で操作が行われる。プレビュー H T M L 生成については、図 3 のプレビュータブ 3 0 9 をユーザが操作したタイミング（プレビュータブ選択 8 3 1）で、当該操作のイベントが、入出力定義エディタコントローラ部 1 0 2 に通知される。 40

【 0 0 7 6 】

入出力定義エディタコントローラ部 1 0 2 は、アプリケーション定義体記憶部のうち、処理に必要な項目一覧情報を取得する（入出力モデル取得 8 3 2）。具体的には、プレビュー H T M L 生成に必要な情報を項目一覧モデル記憶部 1 0 4 から取得する。言わば表示情報のフィルタリングを行う。ここで、図 1 1 を用いて、取得した項目一覧情報について説明する。

【 0 0 7 7 】

図 1 1 は、本発明の実施形態に係わるプレビュー H T M L を生成するために必要な定義 50

情報をアプリケーション定義から取得した結果のデータ構成を示す図の一例である。データ構造としては図5で説明した“rows”とほぼ同様である。まず、開発中のアプリケーションを定義するデータとしては、ビジネスロジックの定義体、データベースモデルの定義体、その他アプリケーションの各種プロパティを定義体など、多くの定義体がある。入出力定義であっても、現在、プレビューHTMLを生成しようとしている入出力定義体と、それ以外の定義体がある。

【0078】

入出力定義エディタコントローラ部102は、まず必要な定義体のみを選択する。具体的には、例えば、図4の項目一覧定義が記載されているデータを含む定義体である。なお、ここで「定義体」という文言は便宜上用いている。実際には、1つの定義体が1ファイルに格納されていても良い。あるいは、1つのアプリケーション全体に関する定義体が1つのファイルに格納され、それぞれの（例えば1つの項目一覧）定義は、タグで判別可能としても良い。この場合は、当該タグの内容を意味する。また、開発中はファイルに格納されておらず、RAM202上にデータが展開されている、あるいはファイルではなくデータベースに格納されていても良い。従って、定義体という文言は、ここでは、1つの項目一覧情報を他の定義情報から区別するための便宜上の用語として使用する。10

【0079】

1つの項目一覧情報を取得したものは、図5のrows500あるいは同等の情報を持つデータ構造で実装される定義体となる。さらに、図4の項目一覧定義においては、“少数”～“データモデル項目コード”などはHTML画面の情報としては不要である。これらは、データベースとの連携を定義するものなどである。そこで、不要な列の削除を行う。図5のrows500で説明すると、例えば2行目の“-1”～“@TEXT”などが不要になるので削除する。これらの処理の結果として、図11のfilteredRows1100が取得される。20

【0080】

ここで、画面情報（ビュー）に関連したビジネスロジック、データベースモデルなどが完成していない段階であっても、プレビューHTMLを生成することが可能であるので、柔軟なタイミングで画面を確認し、効率的な開発を進めるという効果を得ることが可能となる。また、プレビューHTML生成のために必要な情報を用いることで、アプリケーション全体をビルドしなくても画面イメージを確認できることにより、ビルド時間の効率化を得ることができる。30

【0081】

項目一覧情報の取得後、入出力定義エディタコントローラ部102は、プレビューコントローラ部811にプレビューHTML生成要求833を行う。

【0082】

概略として“プレビューテンプレート”という文言を説明すると“プレビューテンプレート”は、項目一覧情報からHTMLのタグの階層構造にマッピングを行うことが可能な、簡易言語であり、プレビューHTML生成部814に保持されている。プレビューテンプレートパラメータ生成部812は、プレビューHTML生成部814に渡すためのプレビューテンプレートパラメータを生成する。40

【0083】

入出力定義エディタコントローラ部102は、まず、プレビューHTMLファイルを生成した後で格納するフルパス名を生成する。ファイル名は、予めユーザにより、プレビューハードドライブ名記憶部821に定義し、プレビューハードドライブ名取得834により取得する。

【0084】

図10は、本発明の実施形態に係わるプレビューハードドライブのファイル保管場所（フォルダ）を指定するユーザ画面を示す図の一例である。このダイアログを表示し、ユーザにHTMLファイルを格納するフォルダ名を入力させる。前記取得したファイル名と合わせ、ファイルのフルパス名が決定する。

【0085】

10

20

30

40

50

さらに、プレビューハイパーテキストを生成するための情報を取得し、データ構成を生成するために、プレビューテンプレートパラメータ生成部812にプレビューテンプレートパラメータ生成要求835を行う。

【0086】

プレビューテンプレートパラメータ生成部812は、プレビューハイパーテキストの生成に必要な情報であるプレビュープロパティを、プレビュープロパティ記憶部822から取得する（プレビュープロパティ取得836。プロパティの詳細については後述する）。さらに入力情報をスタイル情報記憶部823から取得する（スタイル情報取得837。スタイル情報は、ハイパーテキストのスタイルシートなどの情報であり、詳細は後述する）。

【0087】

図9は、本発明の実施形態に係わるプレビューハイパーテキストのプレビューステータスを示す図の一例である。プレビューハイパーテキストタブ900bでは、902にプロパティコードが設定されており、そのプロパティコードに従って、903の各プロパティが読み出され、プレビューハイパーテキストが生成される。プレビューハイパーテキストタブ900aでは、デフォルトのプロパティコード（不図示）を設定している。プレビューハイパーテキストタブ900bのプロパティ903で指定される項目には、例えば“ロール”、（拡張ディレクトリ）“言語”がある（図9の例には更に“表示データディレクトリ”があるが後述）。

【0088】

例えば、“ロール”とは、組織の中の役割（役職や職種）を示すプロパティである。アプリケーションでは、本来同一である画面でも、役職などによって表示する項目の制御をすることがある。従って、プレビューハイパーテキストの生成においても、ロールプロパティを設定することで、当該設定に応じた表示する項目の制御を行い、すなわち、同一の項目一覧定義からあっても、ロールによって異なる画面のプレビューハイパーテキストを生成することを可能にするという効果を得られる。

【0089】

また、（拡張ディレクトリ）“言語”がある。言語プロパティについては、プログラム自動生成ツールが、英語／日本語などの複数の言語に対応している場合に、いずれの言語でプレビューハイパーテキスト画面を生成するかを指定するものである。

【0090】

更に、スタイル情報取得837により、スタイル情報記憶部823から、プレビューハイパーテキストで使用するスタイルシートを取得する。スタイルシートには、ユーザ（開発者）が柔軟に変更可能な部分と、アプリケーション（あるいは、アプリケーション開発を管理するプロジェクト）で固定の部分があっても良い。例えば、文字サイズ、フォント、背景色、線の太さなどの定義である。

【0091】

以上で、ハイパーテキストを生成するためのオブジェクト（入出力フィールド、ボタンなど）の情報は、ハイパーテキストの階層構造を反映する階層的なデータ構成、あるいは階層的なデータ構成と置換可能なデータ構成として格納されている。便宜上、前者の「ハイパーテキストの階層構造を反映する階層的なデータ構成」を前提として説明を継続するが、いずれの形式であっても、最終的にハイパーテキストに変換可能である、同等の情報をもつたデータ構成であればよい。また、前記データ構造は、オブジェクトの配置や名称（例えば、テキストボックスの配置や「ユーザ名」などのタイトル）は情報として含んでいるが、実際の値（例えばユーザ名が「鈴木一郎」などの値）は含んでいないため、以下の説明では便宜上「プレビューテンプレート」と呼ぶことにする（前述の「テンプレート」と同一のもの）。しかしながら、「プレビューテンプレート」は、値（例では「ユーザ名」フィールドに対応する「鈴木一郎」）をどこに代入すればよいか、という情報は、プレビューテンプレートにパラメータ（あるいは変数）として含まれているものとする。

【0092】

ハイパーテキスト画面用に定義した項目一覧を、前述のようなハイパーテキストに変換可能な形式に格納し、後述するハイパーテキストに実際に変換することは、プログラム自動生成ツールなどで周知の

技術であるため詳細の説明は省略する。

【0093】

プレビューテンプレートパラメータ生成部812は、プレビューデータ設定部813に対して、前述の「プレビューテンプレート」にサンプルの値を代入するための、プレビューデータ設定要求838を行う。プレビューデータ設定部813は、代入値データ記憶部824からデータを取得（代入値データ取得839）し、「プレビューテンプレートパラメータ」を作成する。

【0094】

これらの処理結果は、プレビューコントローラ部811に返され、プレビューコントローラ部811は、「プレビューテンプレートパラメータ」に基づきプレビューハテムルを生成するようプレビューハテムル生成部814に要求（テンプレートハテムル変換要求840）する。要求を受けた、841プレビューハテムル生成部は保持するプレビューテンプレートとプレビューテンプレートパラメータをマッピングし、プレビューハテムルを生成する。更に、プレビューハテムル表示要求841により、情報処理装置100の表示装置（開発ツールのプレビュータブ）にプレビューハテムルを表示する（図9の画面）。

10

【0095】

前述の説明では、代入値データ記憶部824により「値が入ったプレビューテンプレートパラメータ」が作成される、と説明したが、プレビューハテムルタブ900aにおいては、「XXXXXX...」などの意味のない値が表示されている（901a）。これは、代入値データ記憶部824に代入するための情報が定義されているか、による。

20

【0096】

具体的には、プロパティ903の“表示データディレクトリ”にフォルダが記載されていない、あるいは記載されていても（例では“C:\\$PreviewData”）、そのフォルダ内にデータが格納されていない場合に、デフォルトとしてプレビューハテムルタブ900aの画面が表示される。これらのデフォルト値は、プログラム中にハードコーディングされていても良いし、デフォルト値記憶部（不図示）に保存されていても良い。デフォルト値であっても、例えば入力フィールドのタイプ（全角文字、半角英数字など）により、複数種類用意していても良い。この場合、項目一覧定義体の各要素において、入力フィールドの定義が明示されている場合にはそのタイプに従う。

【0097】

30

一方、代入値データ記憶部824に代入するための値（例えばCSVファイルなどで用意する）がある場合には、プレビューハテムルタブ900bのように、ユーザに分かりやすい（人間にとて意味のある）値を表示させることが可能である（901b）。

【0098】

以上で、図8のプレビューハテムル生成のための機能構成および各機能の呼び出し関係の説明を完了する。あわせて、図9～図11の説明も完了する。

【0099】

図12は、本発明の実施形態に係わるプレビューハテムルの生成処理を示すフローチャートの一例である。図12のフローチャートの各ステップは、情報処理装置100のCPU201により実行される。なお、機能構成図（図8）の説明において述べた部分については、具体的な詳細事項の説明を省略する場合がある。

40

【0100】

S1201においては、ユーザ操作（101）を、GUI（グラフィカル・ユーザ・インターフェース）における定義画面（プログラム自動生成ツールにおいてプログラムの定義を行う画面）から受け付ける。具体的には、情報処理装置100にて動作しているプログラム自動生成ソフトにおいて、ユーザがプレビュータブ309を選択したイベントを受け付ける。

【0101】

S1202からS1205の繰り返し処理においては、アプリケーション定義情報の全ての定義体から、プレビュー表示要求対象の画面に関連する情報を取得する。

50

S1203においては、アプリケーション定義体記憶部から取得する。

【0102】

S1204においては、S1203で取得した定義体が、プレビュー表示要求対象の項目一覧の定義体であるか判定し、そうであれば繰り返し処理(S1202～S1205)を抜ける。具体的には、当該定義体が、項目一覧モデル記憶部104から取得したものであり、更に、現在定義中(開発中)の項目一覧定義の定義体であるものを取得する。

【0103】

次に、S1206からS1209の繰り返し処理においては、前述で取得した項目一覧の定義体(例えば図5のrows500)から、要素を1つずつ取り出し、全ての要素に対して以下の処理を実行する。

10

【0104】

S1207においては、項目一覧の定義体(例えば図5のrows500)から、要素(row[i])を1つずつ取り出し、前述したとおり、図4の項目一覧定義における“少数”～“データモデル項目コード”などHTML画面の情報としては不要な部分を除き、“項目タイプ”、“項目コード”～“桁数”(図4参照)など、HTML画面のオブジェクトとして必要な情報を抽出する。

【0105】

S1208においては、S1207で抽出した情報を、filteredRows1100に格納する(要素filteredRows[i])。このように、プレビューHTML生成においては、プログラム自動生成ツールで定義した定義体のうち、プレビューHTML生成に必要な情報のみを利用することで、ビジネスロジック、データベースモデルなどが完成していない段階であっても、プレビューHTMLを生成することが可能であり、柔軟なタイミングで画面を確認し、効率的な開発を進めるという効果を得ることが可能となる。また、プレビューHTML生成のために必要な情報を用いることで、アプリケーション全体をビルドしなくても画面イメージを確認できることにより、ビルド時間の効率化を得ることができる。

20

【0106】

S1210においては、プレビューテンプレートを生成し、生成するプレビューHTMLファイルの名前を取得し、プレビューテンプレートに格納する。ここでプレビューテンプレートとは、前述したとおり、HTMLの構造を生成する直前の段階のデータ構成である。即ち、プレビューテンプレートは、HTMLのタグの階層構造にマッピングを行うことで、実際のHTMLを生成可能な情報を含むものである。

30

【0107】

S1211においては、生成するプレビューHTMLファイルのプレビュープロパティを取得し、プレビューテンプレートに格納する。具体的には、図9の903において説明した。

【0108】

S1212は、プレビューHTMLのスタイル情報を取得する。具体的には、スタイルシートの情報である。

40

【0109】

S1213からS1215の繰り返し処理は、filteredRows1100の全要素に対して実行される。

【0110】

S1214においては、filteredRows[i](項目一覧定義の1要素からHTMLの1オブジェクトに対応する表示に必要な情報を取り出したデータ)を、プレビューテンプレートに埋め込んでいく。ただし、S1214の処理が実行されるのは、“ホール”プロパティにより、表示すべきと判断されたfilteredRows[i]のみである。これにより、実質的にはプレビューHTMLと等価なデータ構造ができる。この段階で、フィールド部品に表示するユーザデータには値は入っていない。

【0111】

50

S 1 2 1 6においては、フィールド部品に、具体的な値は入をいれていく。具体的な値は、代入値データ記憶部824から取得する。代入値データ記憶部824には、例えばC S V形式のファイルなどにより、例えばフィールドの項目名（あるいは項目ID）に対応付けて、その入力フィールドに挿入したい値が定義されている。この値を、前述のプレビューテンプレートの変数パと置き換えていく。

【0 1 1 2】

S 1 2 1 7においては、一覧項目の名、型、代入値を、H T M Lオブジェクトに変換し、プレビューH T M Lファイルに埋め込んでいく。

【0 1 1 3】

S 1 2 1 8においては、生成されたプレビューH T M Lファイルを、情報処理装置100の表示装置（開発ツールのプレビュータブ）にプレビューH T M Lを表示する（図9の画面）。

以上で、図12のフローチャートの説明を完了する。

【0 1 1 4】

なお、上述した各種データの構成及びその内容はこれに限定されるものではなく、用途や目的に応じて、様々な構成や内容で構成されることは言うまでもない。

【0 1 1 5】

以上、一実施形態について示したが、本発明は、例えば、システム、装置、方法、プログラムもしくは記録媒体等としての実施態様をとることが可能であり、具体的には、複数の機器から構成されるシステムに適用しても良いし、また、一つの機器からなる装置に適用しても良い。

【0 1 1 6】

また、本発明におけるプログラムは、図6、図7に示すシーケンス図、図12のフローチャートの処理方法をコンピュータが実行可能なプログラムであり、本発明の記録媒体は図6、図7に示すシーケンス図、図12のフローチャートの処理方法をコンピュータが実行可能なプログラムが記憶されている。なお、本発明におけるプログラムは図6、図7、図12のフローチャートに示すシーケンス図の各装置の処理方法ごとのプログラムであつてもよい。

【0 1 1 7】

以上のように、前述した実施形態の機能を実現するプログラムを記録した記録媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはC P UやM P U）が記録媒体に格納されたプログラムを読み出し実行することによっても、本発明の目的が達成されることは言うまでもない。

【0 1 1 8】

この場合、記録媒体から読み出されたプログラム自体が本発明の新規な機能を実現することになり、そのプログラムを記憶した記録媒体は本発明を構成することになる。

【0 1 1 9】

コンピュータプログラムを供給するための記録媒体としては、例えば、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、C D - R O M、C D - R、D V D - R O M、磁気テープ、不揮発性のメモリカード、R O M、E E P R O M、シリコンディスク、ソリッドステートドライブ等を用いることができる。

【0 1 2 0】

また、コンピュータが読み出したプログラムを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムの指示に基づき、コンピュータ上で稼働しているO S（オペレーティングシステム）等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0 1 2 1】

さらに、記録媒体から読み出されたプログラムが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、

10

20

30

40

50

そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるC P U等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【 0 1 2 2 】

また、本発明は、複数の機器から構成されるシステムに適用しても、1つの機器からなる装置に適用してもよい。また、本発明は、システムあるいは装置にプログラムを供給することによって達成される場合にも適応できることは言うまでもない。この場合、本発明を達成するためのプログラムを格納した記録媒体を該システムあるいは装置に読み出すことによって、そのシステムあるいは装置が、本発明の効果を享受することが可能となる。さらに、本発明を達成するためのプログラムをネットワーク上のサーバ、データベース等から通信プログラムによりダウンロードして読み出すことによって、そのシステムあるいは装置が、本発明の効果を享受することが可能となる。10

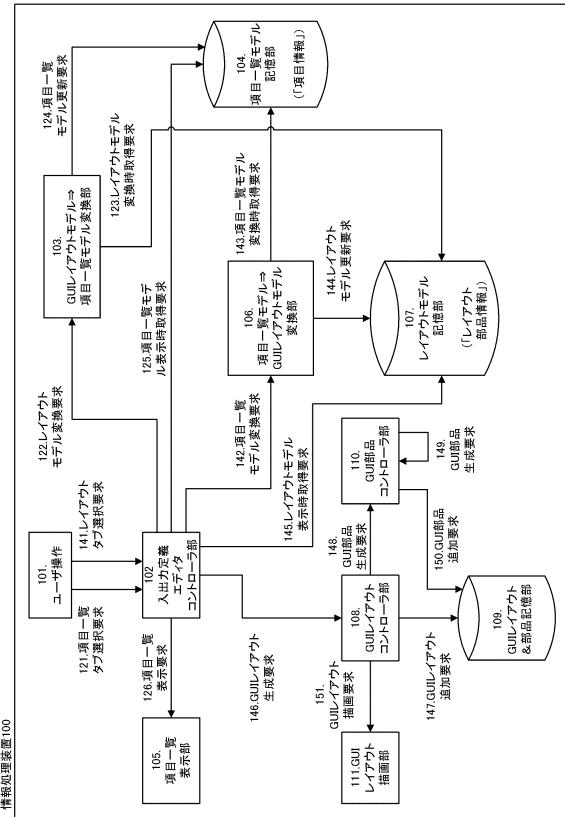
なお、上述した各実施形態およびその変形例を組み合わせた構成も全て本発明に含まれるものである。

【 符号の説明 】

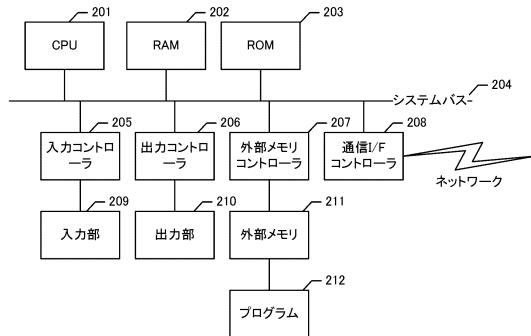
【 0 1 2 3 】

1 0 0	情報処理装置	
1 0 2	入力定義エディタコントローラ部	
1 0 3	項目一覧モデル変換部	
1 0 4	項目一覧モデル記憶部	20
1 0 5	項目一覧表示部	
1 0 6	G U I レイアウトモデル変換部	
1 0 7	レイアウトモデル記憶部	
1 0 8	G U I レイアウトコントローラ部	
1 0 9	G U I レイアウト記憶部	
1 1 0	G U I コントローラ部	
1 1 1	G U I レイアウト描画部	
8 1 1	プレビューコントローラ部	
8 1 2	テンプレート生成部	
8 1 3	プレビューデータ設定部	30
8 1 4	プレビューH T M L生成部	
8 1 5	プレビューH T M L表示部	
8 2 1	プレビューH T M L名記憶部	
8 2 2	プレビュープロパティ記憶部	
8 2 3	プレビュースタイル情報記憶部	
8 2 4	代入値データ記憶部	

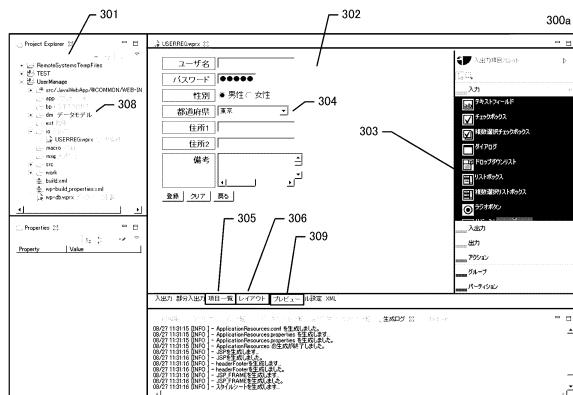
【 図 1 】



【 义 2 】



【図3】



【 四 4 】

【図5】

```

rows[0]
  data: I, USERID, ユーザID, true, false, 1, 20, -1,, @TEXT
  isDirty: false
  rowIndex: 0
  rowInformation
    isDisable: false
  rowPropertyList
    labelWidth: 110
    fieldWidth: 300

rows[1]
  data: IO, USERNAME, ユーザ名, true, false, 1, 20, -1,, USERDM, USERNAME
  isDirty: false
  rowIndex: 1
  rowInformation
    isDisable: false
  rowPropertyList
    labelWidth: 110

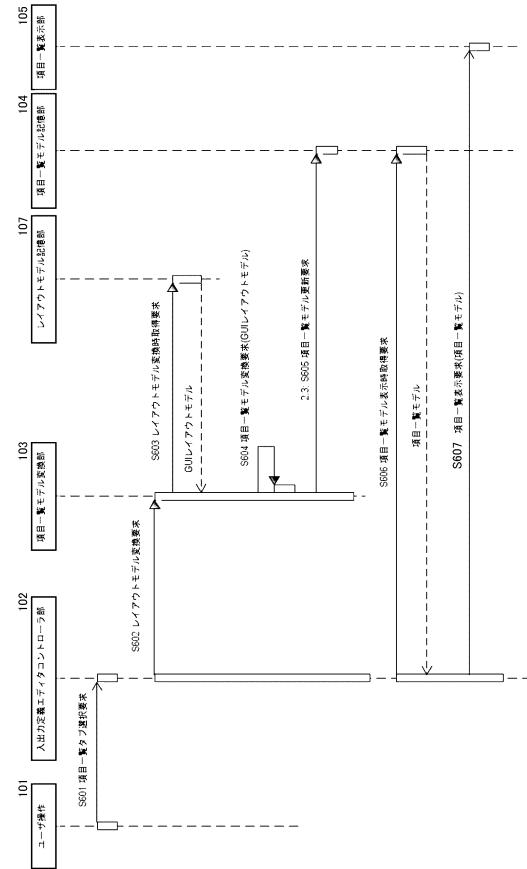
rows[2]
  data: I, PASS, パスワード, true, false, 1, 0, -1,, @TEXT
  isDirty: false
  rowIndex: 2
  rowInformation
    isDisable: false
  rowPropertyList
    labelWidth: 110
    fieldType: PASSWORD

rows[3]
  data: I, SEX, 性別, true, false, 1, 0, -1,, @TEXT
  isDirty: false
  rowIndex: 3
  rowInformation
    isDisable: false
  rowPropertyList
    labelWidth: 110
    fieldType: RADIO

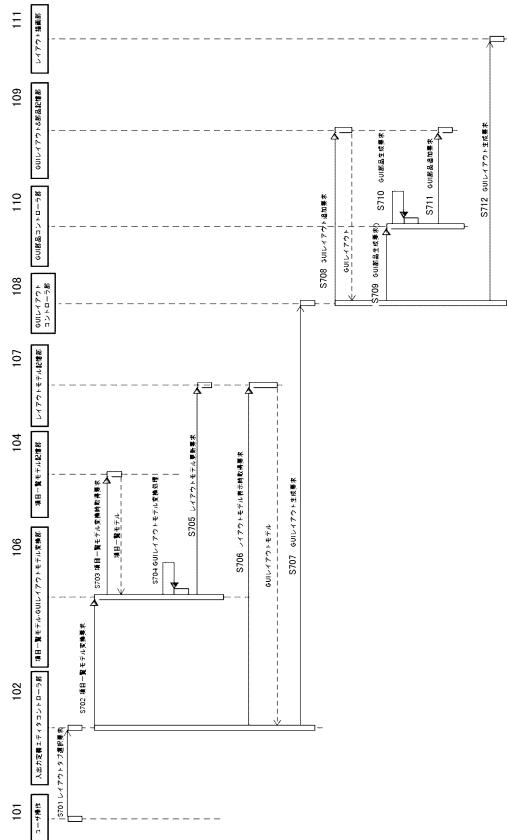
rows[4]
  data: I, TODOFUKEN, 都道府県, true, false, 1, 0, -1,, @TEXT
  isDirty: false
  rowIndex: 4
  rowInformation
    isDisable: false
  rowPropertyList
    labelWidth: 110
    fieldWidth: 128

.
.
```

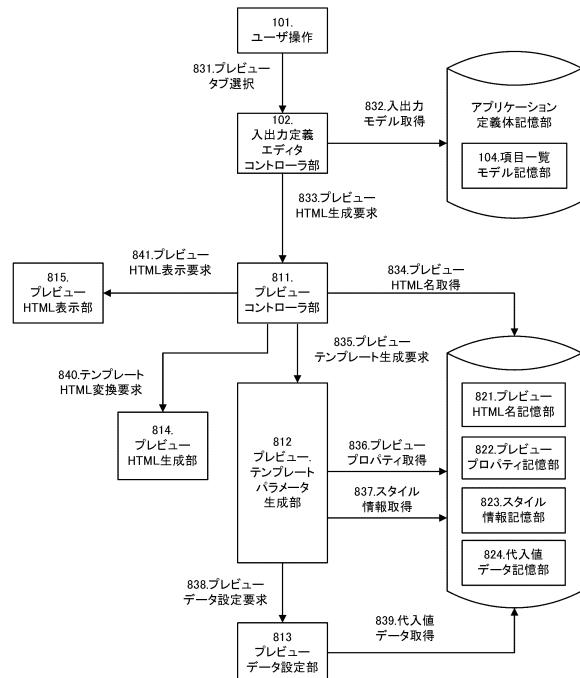
【図6】



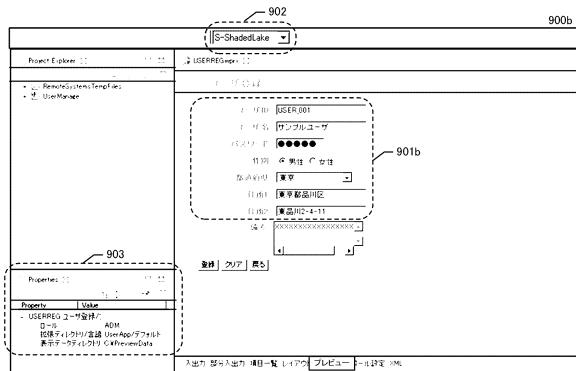
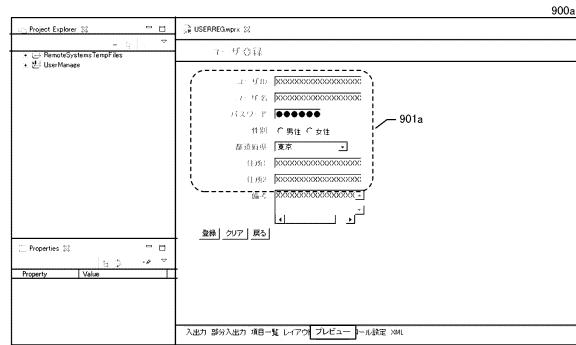
【図7】



【図8】



【図9】



【図10】



【図11】

```
filteredRows[0]
data: I. USERID, ユーザID, true, false, 1, 20
isDirty: false
rowIndex: 0
rowInformation
isDisable: false
rowPropertyList
labelWidth: 110
fieldWidth: 300

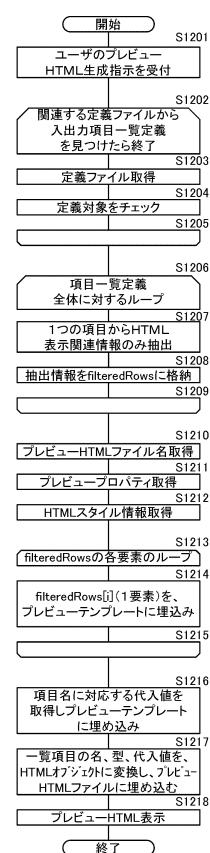
filteredRows[1]
data: I0, USERNAME, ユーザ名, true, false, 1, 20
isDirty: false
rowIndex: 1
rowInformation
isDisable: false
rowPropertyList
labelWidth: 110

filteredRows[2]
data: I, PASS, パスワード, true, false, 1, 0
isDirty: false
rowIndex: 2
rowInformation
isDisable: false
rowPropertyList
labelWidth: 110
fieldType: PASSWORD

filteredRows[3]
data: I, SEX, 性別, true, false, 1, 0
isDirty: false
rowIndex: 3
rowInformation
isDisable: false
rowPropertyList
labelWidth: 110
fieldType: RADIO

filteredRows[4]
data: I, TODOFUKEN, 都道府県, true, false, 1, 0
isDirty: false
rowIndex: 4
rowInformation
isDisable: false
rowPropertyList
labelWidth: 110
fieldWidth: 128
```

【図12】



フロントページの続き

(72)発明者 鈴木 智子

東京都品川区東品川2丁目4番11号 キヤノンソフトウェア株式会社内

(72)発明者 石塚 ゆみ

東京都品川区東品川2丁目4番11号 キヤノンソフトウェア株式会社内

審査官 石川 亮

(56)参考文献 特開2010-073062(JP,A)

特開2008-243105(JP,A)

(58)調査した分野(Int.Cl., DB名)

G 06 F 8 / 38

G 06 F 3 / 0481